

# Jirani App - Quick Reference Guide

Essential commands, troubleshooting, and quick fixes for your Django project.

---

## COMMON COMMANDS

### Start Server

```
bash  
python manage.py runserver
```

Access at: [\(http://127.0.0.1:8000/\)](http://127.0.0.1:8000/)

### Stop Server

Press **Ctrl + C** in terminal

### Run on Different Port

```
bash  
python manage.py runserver 8080
```

## DATABASE COMMANDS

### Create Migration Files

```
bash  
python manage.py makemigrations
```

### Apply Migrations

```
bash  
python manage.py migrate
```

### Create Superuser (Admin)

```
bash  
python manage.py createsuperuser
```

## Open Django Shell

```
bash  
python manage.py shell
```

## View All Migrations

```
bash  
python manage.py showmigrations
```

## Reset Database (DANGER!)

```
bash  
# Delete db.sqlite3  
# Then run:  
python manage.py migrate  
python manage.py createsuperuser
```

## STATIC FILES

### Collect Static Files

```
bash  
python manage.py collectstatic
```

### Clear Collected Static Files

```
bash  
python manage.py collectstatic --clear --noinput
```

## GIT COMMANDS

### Check Status

```
bash  
git status
```

### Add All Changes

```
bash
```

```
git add .
```

## Commit Changes

```
bash
```

```
git commit -m "Your message here"
```

## Push to GitHub

```
bash
```

```
git push origin main
```

## Pull Latest Changes

```
bash
```

```
git pull origin main
```

## View Commit History

```
bash
```

```
git log --oneline
```

## Create New Branch

```
bash
```

```
git checkout -b feature-name
```

## 🔍 TROUBLESHOOTING

### Problem: Page shows "Page not found (404)"

#### Solution 1: Check URL spelling

- ✓ http://127.0.0.1:8000/login/
- ✗ http://127.0.0.1:8000/Login/
- ✗ http://127.0.0.1:8000/login

#### Solution 2: Check urls.py

```
python
```

```
# Make sure path exists  
path('login/', views.login_view, name='login'),
```

---

### Problem: "No module named 'dashboard'"

**Solution:** Add to INSTALLED\_APPS in settings.py

```
python
```

```
INSTALLED_APPS = [  
    ...  
    'dashboard', # Add this  
]
```

---

### Problem: Static files (CSS/JS) not loading

**Solution 1:** Run collectstatic

```
bash
```

```
python manage.py collectstatic
```

**Solution 2:** Check settings.py

```
python
```

```
STATIC_URL = '/static/'  
STATICFILES_DIRS = [BASE_DIR / 'dashboard' / 'static']
```

**Solution 3:** Verify template has

```
html
```

```
{% load static %}  
<link rel="stylesheet" href="{% static 'css/style.css' %}">
```

**Solution 4:** Hard refresh browser

- Windows: Ctrl + F5
  - Mac: Cmd + Shift + R
-

**Problem:** "OperationalError: no such table"

**Solution:** Run migrations

```
bash  
  
python manage.py makemigrations  
python manage.py migrate
```

---

**Problem:** "CSRF verification failed"

**Solution:** Check form has CSRF token

```
html  
  
<form method="POST">  
    {% csrf_token %}  
    ...  
</form>
```

---

**Problem:** Login redirects to /accounts/login/

**Solution:** Add to settings.py

```
python  
  
LOGIN_URL = 'login'  
LOGIN_REDIRECT_URL = 'dashboard'
```

---

**Problem:** "ModuleNotFoundError: No module named 'PIL'"

**Solution:** Install Pillow

```
bash  
  
pip install pillow
```

---

**Problem:** Changes not showing

**Solutions:**

1. Restart server (`(Ctrl+C)` then `(python manage.py runserver)`)
2. Hard refresh browser (`(Ctrl+F5)`)

3. Clear browser cache

4. Check if editing correct file

---

**Problem: "AttributeError: module has no attribute"**

**Solution:** Check imports in views.py

```
python
```

```
from .models import Resident, Payment # Make sure models are imported
```

---

**Problem: Port already in use**

**Solution 1:** Kill process using port

```
bash
```

```
# Windows
```

```
netstat -ano | findstr :8000  
taskkill /PID <PID_NUMBER> /F
```

```
# Mac/Linux
```

```
lsof -ti:8000 | xargs kill -9
```

---

**Solution 2:** Use different port

```
bash
```

```
python manage.py runserver 8080
```

---

## DJANGO SHELL QUICK COMMANDS

**Enter Shell**

```
bash
```

```
python manage.py shell
```

---

**Query All Residents**

```
python
```

```
from dashboard.models import Resident
residents = Resident.objects.all()
for r in residents:
    print(r.unit_number, r.user.username)
```

## Get Specific Resident

```
python

resident = Resident.objects.get(unit_number='294')
print(resident.phone)
```

## Create New Payment

```
python

from dashboard.models import Resident, Payment
from datetime import date

resident = Resident.objects.first()
Payment.objects.create(
    resident=resident,
    amount=5000.00,
    payment_type='rent',
    due_date=date.today()
)
```

## Delete All Payments

```
python

from dashboard.models import Payment
Payment.objects.all().delete()
```

## Count Objects

```
python

from dashboard.models import Resident, Payment
print(f'Residents: {Resident.objects.count()}')
print(f'Payments: {Payment.objects.count()}')
```

## Filter Queries

```
python
```

```
# Get pending requests
from dashboard.models import Request
pending = Request.objects.filter(status='pending')

# Get payments for specific month
from datetime import date
payments = Payment.objects.filter(due_date__month=11)

# Get delayed work orders
from dashboard.models import WorkOrder
delayed = WorkOrder.objects.filter(status='delayed', days_late__gt=2)
```

## Exit Shell

```
python
exit()
```

## ADMIN PANEL SHORTCUTS

### Access Admin Panel

```
http://127.0.0.1:8000/admin/
```

### Register Models in admin.py

```
python
from django.contrib import admin
from .models import Resident, Payment, Request, WorkOrder, Unit

admin.site.register(Resident)
admin.site.register(Payment)
admin.site.register(Request)
admin.site.register(WorkOrder)
admin.site.register(Unit)
```

### Customize Admin Display

```
python
```

```
@admin.register(Resident)
class ResidentAdmin(admin.ModelAdmin):
    list_display = ['unit_number', 'user', 'phone', 'is_active']
    list_filter = ['is_active', 'move_in_date']
    search_fields = ['unit_number', 'user__username', 'phone']
```

## USEFUL DJANGO QUERIES

### Get current user's resident profile

```
python
```

```
resident = Resident.objects.get(user=request.user)
```

### Get all payments for a resident

```
python
```

```
payments = Payment.objects.filter(resident=resident)
```

### Get unpaid payments

```
python
```

```
unpaid = Payment.objects.filter(paid=False)
```

### Get requests ordered by date

```
python
```

```
requests = Request.objects.all().order_by('-created_at')
```

### Count by status

```
python
```

```
from django.db.models import Count
stats = Request.objects.values('status').annotate(count=Count('status'))
```

### Get related objects

```
python
```

```
resident = Resident.objects.get(unit_number='294')
payments = resident.payment_set.all() # All payments for this resident
requests = resident.request_set.all() # All requests for this resident
```

---

## CSS QUICK FIXES

### Center Everything

```
css
```

```
display: flex;  
justify-content: center;  
align-items: center;
```

### Make Responsive

```
css
```

```
@media (max-width: 768px) {  
    .sidebar {  
        display: none;  
    }  
}
```

### Add Gradient

```
css
```

```
background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
```

### Round Corners

```
css
```

```
border-radius: 8px; /* Small */  
border-radius: 12px; /* Medium */  
border-radius: 20px; /* Large */
```

---

## QUICK TIPS

### Restart After Changes

Restart server if you change:

-  models.py
-  settings.py
-  urls.py

- ✗ templates (no restart needed)
- ✗ static files (just refresh browser)

## Debug Mode

In settings.py:

```
python  
  
DEBUG = True # Development (shows errors)  
DEBUG = False # Production (hides errors)
```

## Check Django Version

```
bash  
  
python -m django --version
```

## Check Python Version

```
bash  
  
python --version
```

## Install All Requirements

```
bash  
  
pip install -r requirements.txt
```

## Create Requirements File

```
bash  
  
pip freeze > requirements.txt
```

---

## ⚠️ EMERGENCY FIXES

### Complete Reset (NUCLEAR OPTION)

**Warning:** This deletes ALL data!

```
bash
```

```
# 1. Delete database
rm db.sqlite3 # Mac/Linux
del db.sqlite3 # Windows

# 2. Delete migrations (keep __init__.py)
# Manually delete files in dashboard/migrations/ except __init__.py

# 3. Recreate everything
python manage.py makemigrations
python manage.py migrate
python manage.py createsuperuser

# 4. Add sample data again
python manage.py shell
# Paste sample data script
```

---

## Server Won't Start

### Check for errors:

```
bash
python manage.py check
```

### Common issues:

- Typo in code
- Missing comma
- Incorrect indentation
- Unclosed bracket

**Find the error:** Look at the last line of error message - it tells you the file and line number.

---

## TESTING CHECKLIST

Before presenting:

- Server starts without errors
- Can login successfully
- Dashboard loads completely
- All stats show correct numbers
- Charts display properly

- No broken images
  - CSS is loading
  - JavaScript works (search, animations)
  - Admin panel accessible
  - Responsive on mobile (test with DevTools)
- 

## USEFUL URLs

### Development URLs

```
Main site: http://127.0.0.1:8000/  
Login: http://127.0.0.1:8000/login/  
Dashboard: http://127.0.0.1:8000/  
Admin: http://127.0.0.1:8000/admin/  
Logout: http://127.0.0.1:8000/logout/
```

## DOCUMENTATION LINKS

- Django Docs: <https://docs.djangoproject.com/>
  - Django Tutorial: <https://docs.djangoproject.com/en/stable/intro/tutorial01/>
  - Django ORM Queries: <https://docs.djangoproject.com/en/stable/topics/db/queries/>
  - Django Templates: <https://docs.djangoproject.com/en/stable/topics/templates/>
  - Python Docs: <https://docs.python.org/3/>
- 

## COMMON PATTERNS

### View Pattern

```
python  
  
@login_required  
def my_view(request):  
    # Get data  
    data = Model.objects.all()  
  
    # Process data  
    context = {'data': data}  
  
    # Render template  
    return render(request, 'template.html', context)
```

## URL Pattern

```
python

from django.urls import path
from . import views

urlpatterns = [
    path('page/', views.my_view, name='page'),
]
```

## Template Pattern

```
html

{% load static %}
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="{% static 'css/style.css' %}">
</head>
<body>
    {% for item in data %}
        <div>{{ item.name }}</div>
    {% endfor %}
</body>
</html>
```

## Model Pattern

```
python

class MyModel(models.Model):
    name = models.CharField(max_length=100)
    created_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.name
```

## sos NEED MORE HELP?

1. **Read the error message carefully** - it usually tells you what's wrong
2. **Check Django documentation** - excellent and detailed
3. **Search Stack Overflow** - copy error message

4. Check your code against artifacts I provided

5. Use Django's built-in debugging - set breakpoints with `import pdb; pdb.set_trace()`

---

**Remember:** Every developer faces bugs and errors. It's part of the process! 🍍

You've got this! 🚀